

# working in the bioinformatics lab: NETWORKING (YOU are the HELPDESK )

Manfred Roos<sup>1</sup>

<sup>1</sup>Department of Computer Science  
University Munich

bioinformatics lab prak, 2010

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

# easy firewall with shorewall

- define all connections and directions
- define rules onto ↑
- start/debug
- + automatic rules
- - hundred rules

# easy firewall with shorewall

- define all connections and directions
- define rules onto ↑
- start/debug
- + automatic rules
- - hundred rules

# easy firewall with shorewall

- define all connections and directions
- define rules onto ↑
- start/debug
- + automatic rules
- - hundred rules

# easy firewall with shorewall

- define all connections and directions
- define rules onto ↑
- start/debug
- + automatic rules
- - hundred rules

# easy firewall with shorewall

- define all connections and directions
- define rules onto ↑
- start/debug
- + automatic rules
- - hundred rules

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

# iptables

- work direct on iptables
- performance
- faulty-> no protection @all

# iptables

- work direct on iptables
- performance
- faulty-> no protection @all

# iptables

- work direct on iptables
- performance
- faulty-> no protection @all

# iptables

- `*filter :INPUT DROP [0:0]`
- `-A INPUT -i lo -j ACCEPT`
- `-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT`
- `-A INPUT -m state --state INVALID -j DROP`
- `-A INPUT -s 127.0.0.0/255.0.0.0 ! -i lo -j DROP`

# iptables

- -N SSHBRUTE
- -A SSHBRUTE -m recent --name SSH --set
- -A SSHBRUTE -m recent --name SSH --update --seconds 300 --hitcount 10 -m limit --limit 1/second --limit-burst 100 -j LOG --log-prefix "SSHBRUTE: "
- -A SSHBRUTE -m recent --name SSH --update --seconds 300 --hitcount 10 -j DROP
- -A SSHBRUTE -j ACCEPT
- # Chain for preventing SSH brute-force attacks from off-campus. # Permits 10 new connections within 5 minutes from a single host then drops incoming connections from that host # Note: Beyond a burst of 100 connections we log at up 1 attempt per second to prevent filling of logs

# iptables

- -N SSHBRUTE
- -A SSHBRUTE -m recent --name SSH --set
- -A SSHBRUTE -m recent --name SSH --update --seconds 300 --hitcount 10 -m limit --limit 1/second --limit-burst 100 -j LOG --log-prefix "SSHBRUTE: "
- -A SSHBRUTE -m recent --name SSH --update --seconds 300 --hitcount 10 -j DROP
- -A SSHBRUTE -j ACCEPT
- # Chain for preventing SSH brute-force attacks from off-campus. # Permits 10 new connections within 5 minutes from a single host then drops incoming connections from that host # Note: Beyond a burst of 100 connections we log at up 1 attempt per second to prevent filling of logs

# iptables

- -N ICMPFLOOD
- -A ICMPFLOOD -m recent --set --name ICMP --rsource
- -A ICMPFLOOD -m recent --update --seconds 1 --hitcount 6  
--name ICMP --rsource --rttl -m limit --limit 1/sec --limit-burst 1  
-j LOG --log-prefix "ICMPFLOOD: "
- -A ICMPFLOOD -m recent --update --seconds 1 --hitcount 6  
--name ICMP --rsource --rttl -j DROP
- -A ICMPFLOOD -j ACCEPT
- # Chain for preventing ping flooding - up to 6 pings per second from a single source, again with log limiting # Also prevents us from ICMP REPLY flooding some victim when replying to ICMP ECHO from a spoofed source

# iptables

- -N ICMPFLOOD
- -A ICMPFLOOD -m recent --set --name ICMP --rsource
- -A ICMPFLOOD -m recent --update --seconds 1 --hitcount 6  
--name ICMP --rsource --rttl -m limit --limit 1/sec --limit-burst 1  
-j LOG --log-prefix "ICMPFLOOD: "
- -A ICMPFLOOD -m recent --update --seconds 1 --hitcount 6  
--name ICMP --rsource --rttl -j DROP
- -A ICMPFLOOD -j ACCEPT
- # Chain for preventing ping flooding - up to 6 pings per second from a single source, again with log limiting # Also prevents us from ICMP REPLY flooding some victim when replying to ICMP ECHO from a spoofed source

# iptables

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  
-A FORWARD -m state --state RELATED,ESTABLISHED -j  
ACCEPT  
-A OUTPUT -m state --state RELATED,ESTABLISHED -j  
ACCEPT  
-A INPUT -d 192.168.16.0/24 -j IN ## hostip here  
-A INPUT -s 192.168.16.0/24 -j RE
```

# iptables

```
-A IN -p tcp -m tcp --dport 22,25,80,389,443,465,587,8649 -j  
ACCEPT  
-A IN -p udp -m udp --dport 500,4500,8649 -j ACCEPT  
-A IN -p icmp -m icmp --icmp-type 8 -j ACCEPT  
-A IN -s 192.168.16.0/24 -j RE  
-A IN -j LOGNDROP
```

# iptables

```
-A RE -j LOG --log-prefix "ipt:DROP:" --log-level 6  
-A RE -j REJECT --reject-with icmp-port-unreachable  
-A LOGNDROP -j LOG --log-prefix "ipt:DROP:" --log-level 6  
-A LOGNDROP -j DROP  
-A INPUT DROP  
-A FORWARD DROP  
-A OUTPUT DROP
```

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

- stop flooding
- limit bandwidth
- tunnel/nat
- keep your files/privacy
- stop piracy

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

- check lights!
- check cables/adapters
- ping
- tcpdump /wireshark
- firewall/nmap
- ifconfig /routing
- misc: dhcp, ethernet loops, dns

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

# ssh

- secure
  - public/private key
  - password/without
  - ssh-keygen -t rsa
  - `cat ~/.ssh/id_rsa.pub |ssh user@targethost "cat >> ~/.ssh/authorized_keys"`

# ssh

- secure
- public/private key
- password/without
- ssh-keygen -t rsa
- `cat ~/.ssh/id_rsa.pub |ssh user@targethost "cat >> ~/.ssh/authorized_keys"`

# ssh

- secure
- public/private key
- password/without
- ssh-keygen -t rsa
- `cat ~/.ssh/id_rsa.pub |ssh user@targethost "cat >> ~/.ssh/authorized_keys"`

## ssh

- secure
- public/private key
- password/without
- ssh-keygen -t rsa
- `cat ~/.ssh/id_rsa.pub |ssh user@targethost "cat >> ~/.ssh/authorized_keys"`

# Outline

## 1 Firewall

- easy as in simple
- sophisticated
- what the wall can do

## 2 networking

- no network
- ssh
- IDS

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

- snort
- Samhain
- md5sum/sha1sum
- last -ax
- logrotate
- fail2ban

# Summary

- Security is ...
- non-existent
- try to close biggest holes
- read logfiles

# Summary

- Security is ...
- non-existent
- try to close biggest holes
- read logfiles

# Summary

- Security is ...
- non-existent
- try to close biggest holes
- read logfiles

# For Further Reading I



[http://www.g-loaded.eu/2007/08/10/ssl-enabled-name-based-apache-virtual-hosts-with-mod\\_gnutls/](http://www.g-loaded.eu/2007/08/10/ssl-enabled-name-based-apache-virtual-hosts-with-mod_gnutls/)



<http://www.admin-magazin.de/content/das-snort-ids-erkennt-einbruchsversuche>

